

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-021066

(43)Date of publication of application : 23.01.1998

(51)Int.Cl.

G06F 9/06

G06F 9/06

(21)Application number : 08-173537

(71)Applicant : HITACHI LTD

(22)Date of filing : 03.07.1996

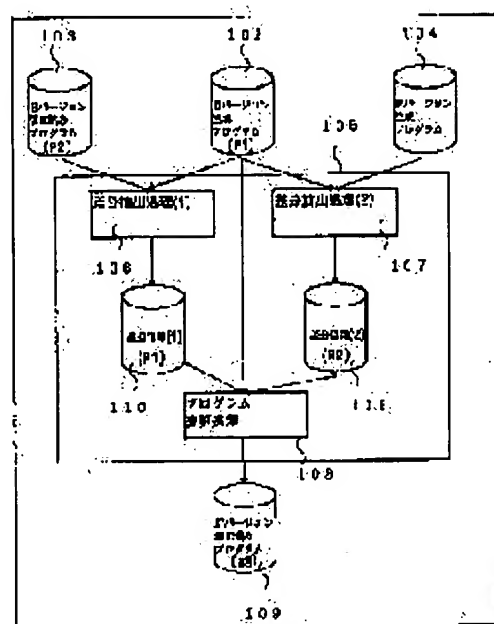
(72)Inventor : KAWAMURA MASAYASU
SAKURAZAWA ASAMI

(54) AUTOMATIC PROGRAM GENERATING METHOD

(57)Abstract:

PROBLEM TO BE SOLVED: To make it possible to use existent properties more effectively and to improve the productivity, by comparing a new version-generating program which has just been generated with specification information after specification alterations with an old version-generating program and reflecting obtained difference information in the old version program.

SOLUTION: The old version-generating program 102 is compared with an old version program 103 which is already corrected to extract difference information (1) 110, the old version-generating program 102 and new version-generating program 104 are further compared with each other to extract difference information (2) 111, and on the basis of those pieces of difference information, the contents of a program to be outputted newly are determined; and the old version-generating program 102 is corrected and outputted as the new version-generating program 109. Consequently, when a system is constituted by diverting the existent program, the existent properties can be used more effectively and the productivity can be improved.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

* NOTICES *

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

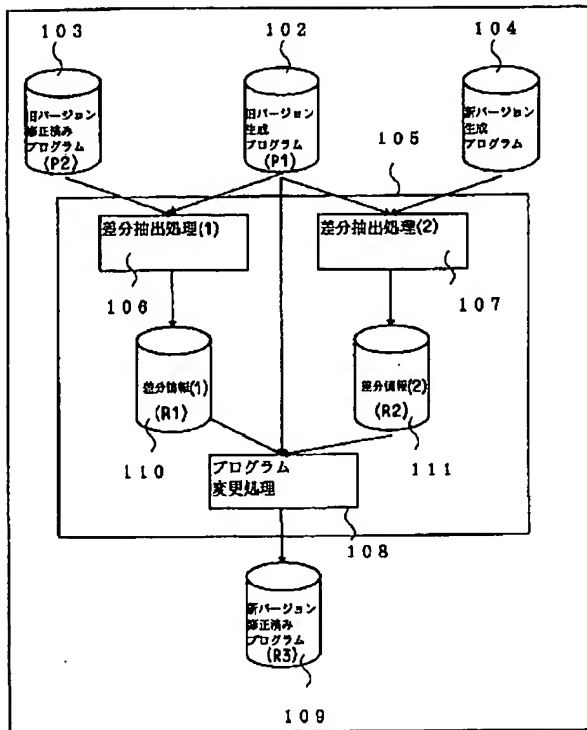
[Claim(s)]

[Claim 1] In the program automatic generation method which develops coding per item to the skeleton of a program from various specification information which is carrying out unitary management of the processing to a data item, such as a data item dictionary and I/O information The 1st step which compares the earlier version generator before making a specification change with the upgrade product generator generated using the upgrade product specification information after specification modification, and extracts the difference, The 2nd step which compares the program and earlier version generator which performed addition and correction / deletion coding to the earlier version generator, and extracts the difference, the difference which these 1st and 2nd means extracted, respectively -- the program automatic generation method reflecting the correction part characterized by having the 3rd step which performs addition, correction, and deletion to the program generated using upgrade product specification information based on information.

[Translation done.]

Drawing selection Representative drawing 

図 1



[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] In the programming process of a software production support system, by holding newly added coding parts, such as deletion, correction, and an addition, to the program which generated automatically, this invention raises the quality of program development and relates to the optimal program automatic generation method for improving working efficiency.

[0002]

[Description of the Prior Art] After adding own coding to the program skeleton which customized and generated the program pattern in the former as proposed by "the programming equipment by pattern use" of JP,4-171532,A, there is the programming approach which made it possible to reflect a required thing in own coding which added the program pattern previously to the program skeleton customized and generated again.

[0003] Also after this program automatic generation technique manages processing (own coding) of a user proper in the place called own-coding Management Department, is characterized by holding additional coding to a skeleton and corrects a skeleton, it can be restored to the corresponding location to the skeleton after correcting own coding. Therefore, it cannot become maintenance of only an additional coding part and the information over the deletion and the correction to the skeleton itself cannot be held.

[0004] Moreover, when looking for the program created using the pattern which corresponds out of the existing program by "the additional coding appropriation method at the time of a programming" of JP,5-100832,A, extracting an additional coding part out of the program and creating a program, the technique [made / the suitable thing / choose and / embed] is proposed from the extracted contents of additional coding to the selected pattern.

[0005] According to this technique, the information on the operating proper processing of those other than the program concerned is acquirable by searching all the existing programs whose patterns used the same thing, and extracting that additional coding part.

[0006] However, you make additional coding to a pattern applicable to an extract, and cannot make it reflected in the pattern which chose this to deletion / correction coding.

[0007]

[Problem(s) to be Solved by the Invention] The above-mentioned conventional program automatic generation method is an effective means which raises the yield at the time of program regeneration, and raises the effectiveness of a programming by holding a user's original processing as an additional coding part, when performing the system development which diverted specification modification and the existing assets.

[0008] However, in generating a program automatically using a basic pattern, the deletion not only to an additional part but creation information and a correction part occur to processing original with a program. For this reason, with the above-mentioned conventional technique, although it can be reflected to additional coding, to deletion / correction coding, there is a problem that it must newly code, at every

regeneration.

[0009] Moreover, even if it applies the program generation method using a data item dictionary or various specification information in business, it is usually very difficult to generate automatically 100%. Processing peculiar to the business which can be judged neither from dictionary information nor definition information is components-ized, or serves as additional coding after automatic generation. However, even if the frequency of occurrence is low and components-izes, low processing of reusability will usually perform additional coding, without components-izing. Moreover, although the program to which mode of processing was similar uses a skeleton in many cases in order to improve productivity, depending on a program, unnecessary logic may occur in this case.

[0010] In the program generation method using a data item dictionary, unitary management of all the processings to a data item is carried out in a data item dictionary. For this reason, two or more processings which will register two or more processings into a data item dictionary when processing of the same data item changes with programs, and were registered into the generate time to one data item will be developed, and unnecessary processing will be deleted after generation. Thus, a certain addition and deletion / correction activity occurs to the generated program. However, with the above-mentioned conventional technique, this deletion and fix information text cannot be held.

[0011] Therefore, whenever it carries out regeneration of the program, there is a problem that correction and deletion of coding must be performed.

[0012] The purpose of this invention is by improving such a trouble and holding the information also about deletion / correction part besides an additional coding part to offer the program automatic generation method reflecting all the correction parts.

[0013]

→ [Means for Solving the Problem] In order to attain the above-mentioned purpose, this invention compares the earlier version generator before making a specification change with the upgrade product generator generated using the upgrade product specification information after specification modification. The 2nd step which compares the 1st step which extracts the difference with the program and earlier version generator which performed addition and correction / deletion coding to the earlier version generator, and extracts the difference, the difference which these 1st and 2nd means extracted, respectively -- it is characterized [main] by having the 3rd step which performs addition, correction, and deletion to the program generated using upgrade product specification information based on information.

[0014] or specification modification did what kind of effect to the earlier version generator by comparing the upgrade product generator immediately after generating for the specification information after the earlier version generator immediately after generating for the specification information on an earlier version, and specification modification according to this invention -- the difference -- information is extracted.

[0015] moreover, the difference which logic to have deleted, corrected and added to the earlier version generator by comparing the earlier version program after performing additional coding after an earlier version generator and generation -- information is extracted.

[0016] these two difference -- making information reflect in an earlier version generator -- an earlier version generator -- the specification information on an upgrade product -- and it remakes to the program in which deletion and correction / addition part was made to reflect.

[0017] Thereby, in the system construction which diverts the existing program, the existing assets can be used more effectively and a yield improves. From this, improvement in the productivity in a programming and a test process and dependability can be aimed at.

[0018]

[Embodiment of the Invention] Hereafter, with reference to a drawing, how to generate the program corrected [upgrade product] automatically is explained to a detail from an earlier version generator, the program corrected [earlier version], and an upgrade product generator.

[0019] Drawing 1 is drawing showing the outline of this invention, inputs into the program automatic generation function 105 the earlier version generator 102 stored in external storage, such as a magnetic

disk, the program 103 corrected [earlier version], and the upgrade product generator 104, and outputs the program 109 corrected [upgrade product].

[0020] the program automatic generation function 105 -- difference -- extract processing (1) 106 and difference -- difference is extracted from each program 102,103,104 using extract processing (2) 107.

[0021] In the earlier version generator 102, it is a program of having generated using the earlier version specification information before making a specification change.

[0022] In the program 103 corrected [earlier version], it is the program which already performed additional coding to the earlier version generator 102.

[0023] In the upgrade product generator 104, it is a program of having generated using the upgrade product specification information after making a specification change.

[0024] In the program 109 corrected [upgrade product], it is the program which the program automatic generation function 105 changed and outputted to the upgrade product generator.

[0025] difference -- extract processing (1) 106 -- the earlier version generator 102 and the program 103 corrected [earlier version] -- an input -- carrying out -- the difference of those programs -- difference -- it is the program outputted as information (1) 110.

[0026] difference -- information (1) 110 will be the information on the difference which described whether it would become the same as that of the program 103 corrected [earlier version], if what kind of correction is added to the earlier version generator 102.

[0027] difference -- extract processing (2) 107 -- the earlier version generator 102 and the upgrade product generator 104 -- comparing -- the difference of those programs -- difference -- it is the program outputted as information (2) 111.

[0028] difference -- the difference which described whether information (2) 111 would become the same as that of the upgrade product generator 104 if what kind of correction is added to the earlier version generator 102 -- it is information.

[0029] the program modification processing 108 -- difference -- information (1) 110 and difference -- information (2) 111 and the earlier version generator 102 -- an input -- carrying out -- difference -- information (1) 110 and difference -- the difference of information (2) 111 -- it is the program which reads the earlier version generator 102, changes the earlier version generator 102 based on information, and is outputted as a program 109 corrected [upgrade product].

[0030] In addition, all these functions are performed on CPU and memory. Moreover, the earlier version generator 102 stored in external storage, such as a magnetic disk, and the upgrade product generator 104 are generable using the program automatic generation method currently indicated by JP,5-108319,A or the approach indicated by the "Hitachi criticism" (March, 1984, vol.66, No.3, p19-p24).

[0031] Next, the procedure of each processing is explained with reference to a flow chart.

[0032] drawing 2 -- difference -- it is the flow chart which shows the procedure of extract processing (1) 106.

[0033] in addition, a drawing -- setting -- an earlier version generator -- P1 and the program corrected [earlier version] -- P2 and difference -- information (1) 110 are written as R1, and are explained.

[0034] First, at step 201, the earlier version generator 102 (P1) is read one line. It judges that it is an end of file at step 202.

[0035] as that in which all the current line of the program 103 corrected [earlier version] or subsequent ones was inserted at step 203 when it was an end of file -- difference -- it outputs as information (1) 110.

[0036] If it is not an end of file, the program 103 corrected [earlier version] will be read one line at step 204. It judges that it is the end of file of the program 103 corrected [earlier version] at step 205.

[0037] as what was deleted at step 206 in all the current line of the earlier version generator 102 or subsequent ones when it was an end of file -- difference -- it outputs as information (1) 110.

[0038] If it is not the end of file of the program 103 corrected [earlier version], step 207 will compare the earlier version generator 102, the program corrected [earlier version], and 103, if the same, it will return to step 201, and if not the same, it will progress to step 208.

[0039] In step 208, the current line number of the earlier version generator 102 is evacuated to a save

area, and the earlier version generator 102 is read one line at the following step 209. And it judges whether it is the end of file of the earlier version generator 102 at the following step 210, if it is an end of file, it will progress to step 213, and if it is not an end of file, it will progress to step 211.

[0040] At step 211, the earlier version generator 102 and the program 103 corrected [earlier version] which were read are compared, if the same, it will progress to step 212, and if not the same, it will return to step 209.

[0041] as what was deleted at step 212 in from the line number evacuated to the save area at step 208 to the current line number of the earlier version generator 102 -- difference -- it outputs as information (1) 110, and progresses to step 214.

[0042] as that in which the program 103 read at step 204 corrected [earlier version] was inserted at step 213 -- difference -- it outputs as information (1) 110, the evacuated line number is recovered, and it progresses to step 214.

[0043] At step 214, the program 103 corrected [earlier version] is read one line, and it progresses to step 205.

[0044] difference -- extract processing (2) 107 extract the difference of the earlier version generator 102 and the upgrade product generator 104 similarly. in addition, the contents of the processing -- difference -- since it is extract processing (1) and the same processing, the explanation by the flow chart is omitted.

[0045] Drawing 3 is a flow chart which shows the contents of processing of the program modification processing 108.

[0046] the following and a drawing -- difference -- information (1) -- 110R1 and difference -- information (2) 111 are written as R2, and are explained.

[0047] first, initiation processing -- setting -- difference -- information (1) 110 -- and a flag -- "0" -- setting -- difference -- information (2) 111 -- and "0" is set to a flag and "0" is set to a save area.

[0048] then, the step 301 -- difference -- information (1) 110 -- and -- if a flag judges that it is "0" and is "0" -- step 302 -- difference -- information (1) 110 of one line are read, and it progresses to step 303, and if it is not "0", it will progress to step 304.

[0049] step 303 -- difference -- if the file of information (1) 110 judges whether it is an end (judging whether it is EOF) and is an end of file -- step 305 -- difference -- information (1) 110 -- and a flag -- "1" -- setting -- the following step 306 -- difference -- it progresses to step 304, after setting "HIGH-VALUE" to the line number of information (1) 110. If it is not an end of file, it will progress to step 304.

[0050] step 304 -- difference -- information (2) 111 -- and -- if a flag "0" comes to judge that it is "0" -- step 307 -- difference -- information (2) 111 of one line are read, and it progresses to step 308, and if it is not "0", it will progress to step 309.

[0051] step 308 -- difference -- if information (2) 111 judge whether it is an end of file and are an end of file -- step 310 -- difference -- information (2) 111 -- and a flag -- "1" -- setting -- further -- step 311 -- difference -- "HIGH-VALUE" is set to the line number of information (2) 111, and it progresses to step 309. If it is not an end of file, it will progress to step 312.

[0052] step 309 -- difference -- if it judges whether the line number of information (2) 111 is the same as a save area and is the same -- step 304 -- return -- if not the same, it will progress to step 312.

[0053] step 312 -- difference -- information (1) 110 -- and a flag -- "1" -- it is -- and difference -- information (2) 111 -- and if a flag judges whether it is "1" and both flags "0" become, a post process will be performed at step 313, and if both the end flag is not "1", it will progress to step 314.

[0054] step 314 -- difference -- the line number of information (1) 110, and difference -- it judges whether the line number of information (2) 111 is the same, if the same, it will progress to step 401 of drawing 4, and if not the same, it will progress to step 315.

[0055] step 315 -- difference -- the line number of information (1) 110 -- difference -- or [being smaller than the line number of information (2) 111] -- it judges whether it is **, if small, it will progress to step 501 of drawing 5, and if not small, it progresses to step 601 of drawing 6.

[0056] At step 401 of drawing 4, a line number is evacuated to a save area, the earlier version generator

102 is read one line at the following step 402, one line is outputted to the program 109 (brief sketch as [A drawing] R3) corrected [upgrade product] at the following step 403, and it progresses to step 404. [0057] step 404 -- difference -- it judges whether the line number of information (1) 110 is the same as the line number of the earlier version generator 102, if the same, it will progress to step 405, and if not the same, it will progress to step 402.

[0058] At step 405, if the difference gestalt of memory 1 (it mentions later by drawing 9) is "insertion", it progresses to step 406, and if it is not "insertion", it will progress to step 407.

[0059] At step 406, the "contents of insertion" of memory 1 is outputted as a program 103 corrected [upgrade product], and it returns to step 301 of drawing 3.

[0060] At step 407, the earlier version generator 102 is read one line, and it returns to step 301 (drawing 3).

[0061] At step 501 of drawing 5, the earlier version generator 102 is read one line, and it progresses to step 502.

[0062] step 502 -- difference -- the line number of information (1) 110 is compared with the line number of the earlier version generator 102, and if the same, it will progress to step 503. When not the same, it progresses to step 507 and the contents of memory 1 are outputted as a program 109 corrected [upgrade product].

[0063] step 503 -- difference -- the contents of the difference gestalt of information (1) 110 distribute processing. When a difference gestalt is "insertion", it progresses to step 504. When a difference gestalt is "deletion", the earlier version generator 102 is read one line at step 505, and it progresses to step 506.

[0064] step 506 -- difference -- information (1) 110 -- one line -- reading -- the following step 508 -- difference -- information (1) 110 judge whether it is an end of file, if it is an end of file, it will progress to step 509, and if there is no end, it will progress to step 312.

[0065] step 509 -- difference -- information (1) 110 -- and "1" is set to a flag and it progresses to step 510. step 510 -- difference -- "HIGH-VALUE" is set to the line number of information (1) 110, and it progresses to step 312.

[0066] Next, at step 601 of drawing 6, the earlier version generator 102 is read one line, and it progresses to step 602.

[0067] step 602 -- difference -- the line number of information (2) 111 is compared with the line number of the earlier version generator 102, and if the same, it will progress to step 603. When not the same, it progresses to step 607 and the contents of memory 1 are outputted to the program 109 corrected [upgrade product].

[0068] step 603 -- difference -- the contents of the difference gestalt of information (2) 111 distribute processing. When a difference gestalt is "insertion", it progresses to step 604. When a difference gestalt is "deletion", at step 605, the earlier version generator 102 is read one line, and is thrown away, and it progresses to step 606.

[0069] step 606 -- difference -- information (2) 111 -- one line -- reading -- step 608 -- difference -- information (2) 111 judge whether it is an end of file, if it is an end of file, it will progress to step 609, and if there is no end, it will return to step 312.

[0070] step 609 -- difference -- information (2) 111 -- and "1" is set to a flag and it progresses to step 610. step 610 -- difference -- "HIGH-VALUE" is set to the line number of information (2) 111, and it returns to step 312.

[0071] drawing 7 -- the example of the earlier version generator 102 and the program 103 corrected [earlier version], and difference -- it is the explanatory view of extract processing (1) 106.

[0072] In the program 103 of this example corrected [earlier version], to the earlier version generator 102, the 2nd line, the 6th line, the 8th line, and the 9th line are inserted, and the 2nd line of the earlier version generator 102 is deleted.

[0073] First, "1" is set to the line number which shows the line under processing about each file of the earlier version generator 102 and the program 103 corrected [earlier version].

[0074] Next, the line "701" which the line number of the earlier version generator 102 points out is read.

[0075] Next, the line "708" which the line number of the program 103 corrected [earlier version] points out is read.

[0076] Next, as a result of comparing "708" with these lines "701", since it is the same, the line number of both the earlier version generator 102 and the program 103 corrected [earlier version] is updated "1" every, namely, a line number is set to "2", and the line "702" which the line number of each file points out, and "709" are read.

[0077] After evacuating the line number of an earlier version generator "102", "2", to a save area 722 since it is not the same as a result of comparing "709" with a line "702", renewal of the line number of the earlier version generator 102 of "1" is carried out, namely, a line number is set to "3", and the line "703" which a line number points out is read. [i.e.,]

[0078] As a result of comparing "709" with a line "703", since it is not the same, the line number of the earlier version generator 102 is "1" Advanced, namely, a line number is set to "4", and the line "704" which a line number points out is read.

[0079] as that "719" in which the line "709" was inserted since it did not become the same but the end of file was reached, although compared by reading a line "705" and "706" similarly hereafter -- difference - - it outputs to information (1) 110.

[0080] Next, the line number of a save area 722, "2", is recovered to the line number of the earlier version generator 102. [i.e.,]

[0081] The line number of the program 103 corrected [earlier version] is "1" Advanced, and it is referred to as "3."

[0082] Next, the line "702" which the line number of the earlier version generator 102 points out is compared with the line "710" which the line number of the program 103 corrected [earlier version] points out. Consequently, since it is the same, perform a line "701", processing about "708", and same processing, and "1" Advance the line number of the earlier version generator 102, and it is referred to as "3", the line number of the program 103 corrected [earlier version] is "1" Advanced, and it is referred to as "4."

[0083] The line "703" which the line number of each file points out, and "711" are read.

[0084] After evacuating the line number of the earlier version generator 102, "3", to a save area 722 since it is not the same as a result of comparing "711" with a line "703", the line number of the earlier version generator 102 is "1" Advanced, it is referred to as "4", the line 704 which the line number of the earlier version generator 102, "4", points out is read, and it compares with "711". [i.e.,] [i.e.,] consequently -- up to the direct continued line of the line [line / which the evacuated line number, "3", points out since it is the same / "703"] which a current line number, "4", points out, i.e., as that "720" from which the line "703" was deleted, -- difference -- it outputs to information (1) 110. [i.e.,] [i.e.,]

[0085] Next, the line number of the earlier version generator 102 is "1" Advanced, and it is referred to as "5", and the line number of the program 103 corrected [earlier version] is "1" Advanced, and it is referred to as "5", and moves to the next processing.

[0086] Next, the line "712" which the line number, "5", of the line "705" and the program 103 corrected [earlier version] which the line number of the earlier version generator 102, "5", points out points out is compared. [i.e.,] [i.e.,] Consequently, since it is the same, a line "701", processing about "708", and same processing are performed, the line number of the earlier version generator 102 is "1" Advanced, and it is referred to as "6", and the line number of the program 103 corrected [earlier version] is "1" Advanced, and it is referred to as "6."

[0087] the same processing as henceforth -- repeating -- a line "721" -- difference -- it outputs to information (1) 110.

[0088] then -- as that "725" by which the line after the line which the line number of the program 103 corrected [earlier version] points out "723", and "724" were inserted in the end of file since the line number of the earlier version generator 102 reached the end of file previously among the line number of the earlier version generator 102, and the line number of the program 103 corrected [earlier version] when a line number was advanced, and "726" -- difference -- it outputs to information (1) 110.

[0089] such processing -- difference -- the result of having performed extract processing -- difference --

to information (1) 110 "MOVE O TO P" (709) is inserted just before the 2nd line "702." "MOVE Q TO R" is inserted for "MOVE C TO J" of the 3rd line "703" deletion and just before the 6th line "706." the difference of an earlier version generator 102 called insertion in "MOVE G TO N" and "MOVE H TO O" to just before an end of file (707), and the program 103 corrected [earlier version] -- information (1) 110 are outputted.

[0090] drawing 8 -- the difference from the earlier version generator 102 and the upgrade product generator 104 -- extract processing (2) 107 -- using -- difference -- the explanatory view having shown the example of processing of extracting -- it is -- the same processing as drawing 7 -- carrying out -- difference -- information (2) 111 are outputted.

[0091] The earlier version generator 102 is received by such processing. "MOVE S TO T" (809) was inserted just before the 2nd (802) line (814), correction of having deleted "MOVE B TO I" of the 2nd (802) line (815) and having deleted "MOVE E TO L" of the 5th (805) line (816) was made -- difference -- it is outputted to information file (2) 111.

[0092] Next, the example of the program modification processing 108 is explained using drawing 9 .

[0093] the program modification processing 108 -- setting -- difference -- information (1) 110 and difference -- information (2) 111 are considered as an input, and it changes by reading the earlier version generator 102, and outputs as a program 109 corrected [upgrade product] .

[0094] difference -- information (1) 110 are the difference of the earlier version generator and the program corrected [earlier version] which were located in a line in order of the line number.

[0095] then, difference -- the line "719" of the 1st line is read from the head of information (1) 110, and it sets to memory (1) 904 as a line number 906, the difference gestalt 907, and contents 908.

[0096] the same -- difference -- the line of the 1st line of information (2) 111 "814" is read, and it sets to memory (2) 905 as a line number 909, the difference gestalt 910, and contents 911.

[0097] Memory (1) "1" If the line number of 904 is compared with the line number of memory (2) 905, since it is the same, the line "701" of the earlier version generator 102 will be written out to the program 109 corrected [upgrade product] as it is, and a line number will be advanced.

[0098] If the line number of the earlier version generator 102 is advanced, since it amounts to the 2nd line, according to "insertion" which is the difference gestalt 907 of memory (1) 904, the contents 908 of insertion of memory (1) 904 will be outputted to the program 109 corrected [upgrade product] with the line number 906 of memory (1) 904, i.e., this example.

[0099] difference -- it the line of the 2nd line information (1) line [of 110] "720" reads, and sets to memory (1) 904.

[0100] the same -- difference -- although the line of the 2nd line of information (2) 111 "815" is read -- a line "815" -- the difference about the 2nd line -- since it is information and the processing about the 2nd line of an earlier version generator is already ended -- again -- difference -- information (2) 111 are read. difference -- the line of the 3rd line of information (2) 111 "816" is read, and it sets to memory (2) 905.

[0101] Memory (1) Since it differs when the line number of 904, i.e., the 3rd line and the line number of memory (2) 905, i.e., the 5th line, is compared, the line number of the smaller one, i.e., the processing about the 3rd line, is performed. The line "702" which the line number of the earlier version generator 110 points out is written out to the program 109 corrected [upgrade product] as it is, and the processing which "1" Advances a line number is repeated until it becomes the line number of the smaller one, i.e., the 3rd line.

[0102] since the line number of the earlier version generator 102 reaches the line number of memory (1) 904, i.e., the 3rd line, -- "deletion" of the difference gestalt 907 of memory (1) 904 -- following -- the line number of the earlier version generator 102 -- "1" -- advancing -- difference -- a line "721" is read from information (1) 110, and it sets to memory (1) 904.

[0103] the same processing as henceforth -- difference -- information (1) 110 and difference -- it continues until both information (2) 111 become an end of file.

[0104] such processing -- difference -- information (1) 110 and difference -- based on the information on information (2) 111, it corrects to the earlier version generator 102, and outputs as lines 927-935 of the program 109 corrected [upgrade product] .

[0105] According to this operation gestalt, the earlier version generator 102 is compared with the program 103 corrected [earlier version] as mentioned above. Extract information (1) 110 and the earlier version generator 102 is further compared with the upgrade product generator 104. difference -- difference -- information (2) 111 -- extracting -- such difference, since the contents of the newly outputted program are determined based on information, the earlier version generator 102 is corrected and it outputs as an upgrade product generator 109 At the time of the system construction which diverted the existing program, the existing assets can be used more effectively, and a yield improves at it. From this, improvement in the productivity and dependability in a programming and a test process can be aimed at.

[0106]

[Effect of the Invention] the difference obtained as mentioned above by comparing the upgrade product generator immediately after generating for the specification information after the earlier version generator immediately after generating this invention for the specification information on an earlier version, and specification modification -- making information reflect in an earlier version generator -- an earlier version generator -- the specification information on an upgrade product -- and it can remake to the program in which deletion and correction / addition part was made to reflect.

[0107] Thereby, the existing assets can be more effectively used at the time of the system construction which diverts the existing program, and a yield improves at it. From this, productivity and improvement in dependability in a programming and a test process can be aimed at.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

TECHNICAL FIELD

[Field of the Invention] In the programming process of a software production support system, by holding newly added coding parts, such as deletion, correction, and an addition, to the program which generated automatically, this invention raises the quality of program development and relates to the optimal program automatic generation method for improving working efficiency.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

PRIOR ART

[Description of the Prior Art] After adding own coding to the program skeleton which customized and generated the program pattern in the former as proposed by "the programming equipment by pattern use" of JP,4-171532,A, there is the programming approach which made it possible to reflect a required thing in own coding which added the program pattern previously to the program skeleton customized and generated again.

[0003] Also after this program automatic generation technique manages processing (own coding) of a user proper in the place called own-coding Management Department, is characterized by holding additional coding to a skeleton and corrects a skeleton, it can be restored to the corresponding location to the skeleton after correcting own coding. Therefore, it cannot become maintenance of only an additional coding part and the information over the deletion and the correction to the skeleton itself cannot be held.

[0004] Moreover, when looking for the program created using the pattern which corresponds out of the existing program by "the additional coding appropriation method at the time of a programming" of JP,5-100832,A, extracting an additional coding part out of the program and creating a program, the technique [made / the suitable thing / choose and / embed] is proposed from the extracted contents of additional coding to the selected pattern.

[0005] According to this technique, the information on the operating proper processing of those other than the program concerned is acquirable by searching all the existing programs whose patterns used the same thing, and extracting that additional coding part.

[0006] However, you make additional coding to a pattern applicable to an extract, and cannot make it reflected in the pattern which chose this to deletion / correction coding.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

EFFECT OF THE INVENTION

[Effect of the Invention] the difference obtained as mentioned above by comparing the upgrade product generator immediately after generating for the specification information after the earlier version generator immediately after generating this invention for the specification information on an earlier version, and specification modification -- making information reflect in an earlier version generator -- an earlier version generator -- the specification information on an upgrade product -- and it can remake to the program in which deletion and correction / addition part was made to reflect.

[0107] Thereby, the existing assets can be more effectively used at the time of the system construction which diverts the existing program, and a yield improves at it. From this, productivity and improvement in dependability in a programming and a test process can be aimed at.

[Translation done.]

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] The above-mentioned conventional program automatic generation method is an effective means which raises the yield at the time of program regeneration, and raises the effectiveness of a programming by holding a user's original processing as an additional coding part, when performing the system development which diverted specification modification and the existing assets.

[0008] However, in generating a program automatically using a basic pattern, the deletion not only to an additional part but creation information and a correction part occur to processing original with a program. For this reason, with the above-mentioned conventional technique, although it can be reflected to additional coding, to deletion / correction coding, there is a problem that it must newly code, at every regeneration.

[0009] Moreover, even if it applies the program generation method using a data item dictionary or various specification information in business, it is usually very difficult to generate automatically 100%. Processing peculiar to the business which can be judged neither from dictionary information nor definition information is components-sized, or serves as additional coding after automatic generation. However, even if the frequency of occurrence is low and components-sizes, low processing of reusability will usually perform additional coding, without components-izing. Moreover, although the program to which mode of processing was similar uses a skeleton in many cases in order to improve productivity, depending on a program, unnecessary logic may occur in this case.

[0010] In the program generation method using a data item dictionary, unitary management of all the processings to a data item is carried out in a data item dictionary. For this reason, two or more processings which will register two or more processings into a data item dictionary when processing of the same data item changes with programs, and were registered into the generate time to one data item will be developed, and unnecessary processing will be deleted after generation. Thus, a certain addition and deletion / correction activity occurs to the generated program. However, with the above-mentioned conventional technique, this deletion and fix information text cannot be held.

[0011] Therefore, whenever it carries out regeneration of the program, there is a problem that correction and deletion of coding must be performed.

[0012] The purpose of this invention is by improving such a trouble and holding the information also about deletion / correction part besides an additional coding part to offer the program automatic generation method reflecting all the correction parts.

[Translation done.]

JAPANESE

[JP,10-021066,A]

CLAIMS DETAILED DESCRIPTION TECHNICAL FIELD PRIOR ART EFFECT OF THE
INVENTION TECHNICAL PROBLEM MEANS DESCRIPTION OF DRAWINGS DRAWINGS

[Translation done.]

* NOTICES *

JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

MEANS

[Means for Solving the Problem] In order to attain the above-mentioned purpose, this invention compares the earlier version generator before making a specification change with the upgrade product generator generated using the upgrade product specification information after specification modification. The 2nd step which compares the 1st step which extracts the difference with the program and earlier version generator which performed addition and correction / deletion coding to the earlier version generator, and extracts the difference, the difference which these 1st and 2nd means extracted, respectively -- it is characterized [main] by having the 3rd step which performs addition, correction, and deletion to the program generated using upgrade product specification information based on information.

[0014] or specification modification did what kind of effect to the earlier version generator by comparing the upgrade product generator immediately after generating for the specification information after the earlier version generator immediately after generating for the specification information on an earlier version, and specification modification according to this invention -- the difference -- information is extracted.

[0015] moreover, the difference which logic to have deleted, corrected and added to the earlier version generator by comparing the earlier version program after performing additional coding after an earlier version generator and generation -- information is extracted.

[0016] these two difference -- making information reflect in an earlier version generator -- an earlier version generator -- the specification information on an upgrade product -- and it remakes to the program in which deletion and correction / addition part was made to reflect.

[0017] Thereby, in the system construction which diverts the existing program, the existing assets can be used more effectively and a yield improves. From this, improvement in the productivity in a programming and a test process and dependability can be aimed at.

[0018]

[Embodiment of the Invention] Hereafter, with reference to a drawing, how to generate the program corrected [upgrade product] automatically is explained to a detail from an earlier version generator, the program corrected [earlier version], and an upgrade product generator.

[0019] Drawing 1 is drawing showing the outline of this invention, inputs into the program automatic generation function 105 the earlier version generator 102 stored in external storage, such as a magnetic disk, the program 103 corrected [earlier version], and the upgrade product generator 104, and outputs the program 109 corrected [upgrade product].

[0020] the program automatic generation function 105 -- difference -- extract processing (1) 106 and difference -- difference is extracted from each program 102, 103, 104 using extract processing (2) 107.

[0021] In the earlier version generator 102, it is a program of having generated using the earlier version specification information before making a specification change.

[0022] In the program 103 corrected [earlier version], it is the program which already performed additional coding to the earlier version generator 102.

[0023] In the upgrade product generator 104, it is a program of having generated using the upgrade

product specification information after making a specification change.

[0024] In the program 109 corrected [upgrade product], it is the program which the program automatic generation function 105 changed and outputted to the upgrade product generator.

[0025] difference -- extract processing (1) 106 -- the earlier version generator 102 and the program 103 corrected [earlier version] -- an input -- carrying out -- the difference of those programs -- difference -- it is the program outputted as information (1) 110.

[0026] difference -- information (1) 110 will be the information on the difference which described whether it would become the same as that of the program 103 corrected [earlier version], if what kind of correction is added to the earlier version generator 102.

[0027] difference -- extract processing (2) 107 -- the earlier version generator 102 and the upgrade product generator 104 -- comparing -- the difference of those programs -- difference -- it is the program outputted as information (2) 111.

[0028] difference -- the difference which described whether information (2) 111 would become the same as that of the upgrade product generator 104 if what kind of correction is added to the earlier version generator 102 -- it is information.

[0029] the program modification processing 108 -- difference -- information (1) 110 and difference -- information (2) 111 and the earlier version generator 102 -- an input -- carrying out -- difference -- information (1) 110 and difference -- the difference of information (2) 111 -- it is the program which reads the earlier version generator 102, changes the earlier version generator 102 based on information, and is outputted as a program 109 corrected [upgrade product].

[0030] In addition, all these functions are performed on CPU and memory. Moreover, the earlier version generator 102 stored in external storage, such as a magnetic disk, and the upgrade product generator 104 are generable using the program automatic generation method currently indicated by JP,5-108319,A or the approach indicated by the "Hitachi criticism" (March, 1984, vol.66, No.3, p19-p24).

[0031] Next, the procedure of each processing is explained with reference to a flow chart.

[0032] drawing 2 -- difference -- it is the flow chart which shows the procedure of extract processing (1) 106.

[0033] in addition, a drawing -- setting -- an earlier version generator -- P1 and the program corrected [earlier version] -- P2 and difference -- information (1) 110 are written as R1, and are explained.

[0034] First, at step 201, the earlier version generator 102 (P1) is read one line. It judges that it is an end of file at step 202.

[0035] as that in which all the current line of the program 103 corrected [earlier version] or subsequent ones was inserted at step 203 when it was an end of file -- difference -- it outputs as information (1) 110.

[0036] If it is not an end of file, the program 103 corrected [earlier version] will be read one line at step 204. It judges that it is the end of file of the program 103 corrected [earlier version] at step 205.

[0037] as what was deleted at step 206 in all the current line of the earlier version generator 102 or subsequent ones when it was an end of file -- difference -- it outputs as information (1) 110.

[0038] If it is not the end of file of the program 103 corrected [earlier version], step 207 will compare the earlier version generator 102, the program corrected [earlier version], and 103, if the same, it will return to step 201, and if not the same, it will progress to step 208.

[0039] In step 208, the current line number of the earlier version generator 102 is evacuated to a save area, and the earlier version generator 102 is read one line at the following step 209. And it judges whether it is the end of file of the earlier version generator 102 at the following step 210, if it is an end of file, it will progress to step 213, and if it is not an end of file, it will progress to step 211.

[0040] At step 211, the earlier version generator 102 and the program 103 corrected [earlier version] which were read are compared, if the same, it will progress to step 212, and if not the same, it will return to step 209.

[0041] as what was deleted at step 212 in from the line number evacuated to the save area at step 208 to the current line number of the earlier version generator 102 -- difference -- it outputs as information (1) 110, and progresses to step 214.

[0042] as that in which the program 103 read at step 204 corrected [earlier version] was inserted at step 213 -- difference -- it outputs as information (1) 110, the evacuated line number is recovered, and it progresses to step 214.

[0043] At step 214, the program 103 corrected [earlier version] is read one line, and it progresses to step 205.

[0044] difference -- extract processing (2) 107 extract the difference of the earlier version generator 102 and the upgrade product generator 104 similarly. in addition, the contents of the processing -- difference -- since it is extract processing (1) and the same processing, the explanation by the flow chart is omitted.

[0045] Drawing 3 is a flow chart which shows the contents of processing of the program modification processing 108.

[0046] the following and a drawing -- difference -- information (1) -- 110R1 and difference -- information (2) 111 are written as R2, and are explained.

[0047] first, initiation processing -- setting -- difference -- information (1) 110 -- and a flag -- "0" -- setting -- difference -- information (2) 111 -- and "0" is set to a flag and "0" is set to a save area.

[0048] then, the step 301 -- difference -- information (1) 110 -- and -- if a flag judges that it is "0" and is "0" -- step 302 -- difference -- information (1) 110 of one line are read, and it progresses to step 303, and if it is not "0", it will progress to step 304.

[0049] step 303 -- difference -- if the file of information (1) 110 judges whether it is an end (judging whether it is EOF) and is an end of file -- step 305 -- difference -- information (1) 110 -- and a flag -- "1" -- setting -- the following step 306 -- difference -- it progresses to step 304, after setting "HIGH-VALUE" to the line number of information (1) 110. If it is not an end of file, it will progress to step 304.

[0050] step 304 -- difference -- information (2) 111 -- and -- if a flag "0" comes to judge that it is "0" -- step 307 -- difference -- information (2) 111 of one line are read, and it progresses to step 308, and if it is not "0", it will progress to step 309.

[0051] step 308 -- difference -- if information (2) 111 judge whether it is an end of file and are an end of file -- step 310 -- difference -- information (2) 111 -- and a flag -- "1" -- setting -- further -- step 311 -- difference -- "HIGH-VALUE" is set to the line number of information (2) 111, and it progresses to step 309. If it is not an end of file, it will progress to step 312.

[0052] step 309 -- difference -- if it judges whether the line number of information (2) 111 is the same as a save area and is the same -- step 304 -- return -- if not the same, it will progress to step 312.

[0053] step 312 -- difference -- information (1) 110 -- and a flag -- "1" -- it is -- and difference -- information (2) 111 -- and if a flag judges whether it is "1" and both flags "0" become, a post process will be performed at step 313, and if both the end flag is not "1", it will progress to step 314.

[0054] step 314 -- difference -- the line number of information (1) 110, and difference -- it judges whether the line number of information (2) 111 is the same, if the same, it will progress to step 401 of drawing 4, and if not the same, it will progress to step 315.

[0055] step 315 -- difference -- the line number of information (1) 110 -- difference -- or [being smaller than the line number of information (2) 111] -- it judges whether it is **, if small, it will progress to step 501 of drawing 5, and if not small, it progresses to step 601 of drawing 6.

[0056] At step 401 of drawing 4, a line number is evacuated to a save area, the earlier version generator 102 is read one line at the following step 402, one line is outputted to the program 109 (brief sketch as [A drawing] R3) corrected [upgrade product] at the following step 403, and it progresses to step 404.

[0057] step 404 -- difference -- it judges whether the line number of information (1) 110 is the same as the line number of the earlier version generator 102, if the same, it will progress to step 405, and if not the same, it will progress to step 402.

[0058] At step 405, if the difference gestalt of memory 1 (it mentions later by drawing 9) is "insertion", it progresses to step 406, and if it is not "insertion", it will progress to step 407.

[0059] At step 406, the "contents of insertion" of memory 1 is outputted as a program 103 corrected [upgrade product], and it returns to step 301 of drawing 3.

[0060] At step 407, the earlier version generator 102 is read one line, and it returns to step 301 (drawing 3).

[0061] At step 501 of drawing 5 , the earlier version generator 102 is read one line, and it progresses to step 502.

[0062] step 502 -- difference -- the line number of information (1) 110 is compared with the line number of the earlier version generator 102, and if the same, it will progress to step 503. When not the same, it progresses to step 507 and the contents of memory 1 are outputted as a program 109 corrected [upgrade product].

[0063] step 503 -- difference -- the contents of the difference gestalt of information (1) 110 distribute processing. When a difference gestalt is "insertion", it progresses to step 504. When a difference gestalt is "deletion", the earlier version generator 102 is read one line at step 505, and it progresses to step 506.

[0064] step 506 -- difference -- information (1) 110 -- one line -- reading -- the following step 508 -- difference -- information (1) 110 judge whether it is an end of file, if it is an end of file, it will progress to step 509, and if there is no end, it will progress to step 312.

[0065] step 509 -- difference -- information (1) 110 -- and "1" is set to a flag and it progresses to step 510. step 510 -- difference -- "HIGH-VALUE" is set to the line number of information (1) 110, and it progresses to step 312.

[0066] Next, at step 601 of drawing 6 , the earlier version generator 102 is read one line, and it progresses to step 602.

[0067] step 602 -- difference -- the line number of information (2) 111 is compared with the line number of the earlier version generator 102, and if the same, it will progress to step 603. When not the same, it progresses to step 607 and the contents of memory 1 are outputted to the program 109 corrected [upgrade product].

[0068] step 603 -- difference -- the contents of the difference gestalt of information (2) 111 distribute processing. When a difference gestalt is "insertion", it progresses to step 604. When a difference gestalt is "deletion", at step 605, the earlier version generator 102 is read one line, and is thrown away, and it progresses to step 606.

[0069] step 606 -- difference -- information (2) 111 -- one line -- reading -- step 608 -- difference -- information (2) 111 judge whether it is an end of file, if it is an end of file, it will progress to step 609, and if there is no end, it will return to step 312.

[0070] step 609 -- difference -- information (2) 111 -- and "1" is set to a flag and it progresses to step 610. step 610 -- difference -- "HIGH-VALUE" is set to the line number of information (2) 111, and it returns to step 312.

[0071] drawing 7 -- the example of the earlier version generator 102 and the program 103 corrected [earlier version], and difference -- it is the explanatory view of extract processing (1) 106.

[0072] In the program 103 of this example corrected [earlier version], to the earlier version generator 102, the 2nd line, the 6th line, the 8th line, and the 9th line are inserted, and the 2nd line of the earlier version generator 102 is deleted.

[0073] First, "1" is set to the line number which shows the line under processing about each file of the earlier version generator 102 and the program 103 corrected [earlier version].

[0074] Next, the line "701" which the line number of the earlier version generator 102 points out is read.

[0075] Next, the line "708" which the line number of the program 103 corrected [earlier version] points out is read.

[0076] Next, as a result of comparing "708" with these lines "701", since it is the same, the line number of both the earlier version generator 102 and the program 103 corrected [earlier version] is updated "1" every, namely, a line number is set to "2", and the line "702" which the line number of each file points out, and "709" are read.

[0077] After evacuating the line number of an earlier version generator "102", "2", to a save area 722 since it is not the same as a result of comparing "709" with a line "702", renewal of the line number of the earlier version generator 102 of "1" is carried out, namely, a line number is set to "3", and the line

"703" which a line number points out is read. [i.e.,]

[0078] As a result of comparing "709" with a line "703", since it is not the same, the line number of the earlier version generator 102 is "1" Advanced, namely, a line number is set to "4", and the line "704" which a line number points out is read.

[0079] as that "719" in which the line "709" was inserted since it did not become the same but the end of file was reached, although compared by reading a line "705" and "706" similarly hereafter -- difference - - it outputs to information (1) 110.

[0080] Next, the line number of a save area 722, "2", is recovered to the line number of the earlier version generator 102. [i.e.,]

[0081] The line number of the program 103 corrected [earlier version] is "1" Advanced, and it is referred to as "3."

[0082] Next, the line "702" which the line number of the earlier version generator 102 points out is compared with the line "710" which the line number of the program 103 corrected [earlier version] points out. Consequently, since it is the same, perform a line "701", processing about "708", and same processing, and "1" Advance the line number of the earlier version generator 102, and it is referred to as "3", the line number of the program 103 corrected [earlier version] is "1" Advanced, and it is referred to as "4."

[0083] The line "703" which the line number of each file points out, and "711" are read.

[0084] After evacuating the line number of the earlier version generator 102, "3", to a save area 722 since it is not the same as a result of comparing "711" with a line "703", the line number of the earlier version generator 102 is "1" Advanced, it is referred to as "4", the line 704 which the line number of the earlier version generator 102, "4", points out is read, and it compares with "711". [i.e.,] [i.e.,] consequently -- up to the direct continued line of the line [line / which the evacuated line number, "3", points out since it is the same / "703"] which a current line number, "4", points out, i.e., as that "720" from which the line "703" was deleted, -- difference -- it outputs to information (1) 110. [i.e.,] [i.e.,]

[0085] Next, the line number of the earlier version generator 102 is "1" Advanced, and it is referred to as "5", and the line number of the program 103 corrected [earlier version] is "1" Advanced, and it is referred to as "5", and moves to the next processing.

[0086] Next, the line "712" which the line number, "5", of the line "705" and the program 103 corrected [earlier version] which the line number of the earlier version generator 102, "5", points out points out is compared. [i.e.,] [i.e.,] Consequently, since it is the same, a line "701", processing about "708", and same processing are performed, the line number of the earlier version generator 102 is "1" Advanced, and it is referred to as "6", and the line number of the program 103 corrected [earlier version] is "1" Advanced, and it is referred to as "6."

[0087] the same processing as henceforth -- repeating -- a line "721" -- difference -- it outputs to information (1) 110.

[0088] then -- as that "725" by which the line after the line which the line number of the program 103 corrected [earlier version] points out "723", and "724" were inserted in the end of file since the line number of the earlier version generator 102 reached the end of file previously among the line number of the earlier version generator 102, and the line number of the program 103 corrected [earlier version] when a line number was advanced, and "726" -- difference -- it outputs to information (1) 110.

[0089] such processing -- difference -- the result of having performed extract processing -- difference -- to information (1) 110 "MOVE O TO P" (709) is inserted just before the 2nd line "702." "MOVE Q TO R" is inserted for "MOVE C TO J" of the 3rd line "703" deletion and just before the 6th line "706." the difference of an earlier version generator 102 called insertion in "MOVE G TO N" and "MOVE H TO O" to just before an end of file (707), and the program 103 corrected [earlier version] -- information (1) 110 are outputted.

[0090] drawing 8 -- the difference from the earlier version generator 102 and the upgrade product generator 104 -- extract processing (2) 107 -- using -- difference -- the explanatory view having shown the example of processing of extracting -- it is -- the same processing as drawing 7 -- carrying out -- difference -- information (2) 111 are outputted.

[0091] The earlier version generator 102 is received by such processing. "MOVE S TO T" (809) was inserted just before the 2nd (802) line (814), correction of having deleted "MOVE B TO I" of the 2nd (802) line (815) and having deleted "MOVE E TO L" of the 5th (805) line (816) was made -- difference -- it is outputted to information file (2) 111.

[0092] Next, the example of the program modification processing 108 is explained using drawing 9.

[0093] the program modification processing 108 -- setting -- difference -- information (1) 110 and difference -- information (2) 111 are considered as an input, and it changes by reading the earlier version generator 102, and outputs as a program 109 corrected [upgrade product].

[0094] difference -- information (1) 110 are the difference of the earlier version generator and the program corrected [earlier version] which were located in a line in order of the line number.

[0095] then, difference -- the line "719" of the 1st line is read from the head of information (1) 110, and it sets to memory (1) 904 as a line number 906, the difference gestalt 907, and contents 908.

[0096] the same -- difference -- the line of the 1st line of information (2) 111 "814" is read, and it sets to memory (2) 905 as a line number 909, the difference gestalt 910, and contents 911.

[0097] Memory (1) "1" If the line number of 904 is compared with the line number of memory (2) 905, since it is the same, the line "701" of the earlier version generator 102 will be written out to the program 109 corrected [upgrade product] as it is, and a line number will be advanced.

[0098] If the line number of the earlier version generator 102 is advanced, since it amounts to the 2nd line, according to "insertion" which is the difference gestalt 907 of memory (1) 904, the contents 908 of insertion of memory (1) 904 will be outputted to the program 109 corrected [upgrade product] with the line number 906 of memory (1) 904, i.e., this example.

[0099] difference -- it the line of the 2nd line information (1) line [of 110] "720" reads, and sets to memory (1) 904.

[0100] the same -- difference -- although the line of the 2nd line of information (2) 111 "815" is read -- a line "815" -- the difference about the 2nd line -- since it is information and the processing about the 2nd line of an earlier version generator is already ended -- again -- difference -- information (2) 111 are read. difference -- the line of the 3rd line of information (2) 111 "816" is read, and it sets to memory (2) 905.

[0101] Memory (1) Since it differs when the line number of 904, i.e., the 3rd line and the line number of memory (2) 905, i.e., the 5th line, is compared, the line number of the smaller one, i.e., the processing about the 3rd line, is performed. The line "702" which the line number of the earlier version generator 110 points out is written out to the program 109 corrected [upgrade product] as it is, and the processing which "1" Advances a line number is repeated until it becomes the line number of the smaller one, i.e., the 3rd line.

[0102] since the line number of the earlier version generator 102 reaches the line number of memory (1) 904, i.e., the 3rd line, -- "deletion" of the difference gestalt 907 of memory (1) 904 -- following -- the line number of the earlier version generator 102 -- "1" -- advancing -- difference -- a line "721" is read from information (1) 110, and it sets to memory (1) 904.

[0103] the same processing as henceforth -- difference -- information (1) 110 and difference -- it continues until both information (2) 111 become an end of file.

[0104] such processing -- difference -- information (1) 110 and difference -- based on the information on information (2) 111, it corrects to the earlier version generator 102, and outputs as lines 927-935 of the program 109 corrected [upgrade product].

[0105] According to this operation gestalt, the earlier version generator 102 is compared with the program 103 corrected [earlier version] as mentioned above. Extract information (1) 110 and the earlier version generator 102 is further compared with the upgrade product generator 104. difference -- difference -- information (2) 111 -- extracting -- such difference, since the contents of the newly outputted program are determined based on information, the earlier version generator 102 is corrected and it outputs as an upgrade product generator 109 At the time of the system construction which diverted the existing program, the existing assets can be used more effectively, and a yield improves at it. From this, improvement in the productivity and dependability in a programming and a test process can be aimed at.

* NOTICES *

JPO and NCIPi are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the explanatory view showing the outline of the procedure of this invention.

[Drawing 2] It is the flow chart which shows the processing which extracts the difference of an earlier version generator and the program corrected [earlier version].

[Drawing 3] It is the flow chart which shows program modification processing.

[Drawing 4] It is the flow chart which shows a continuation of drawing 3 .

[Drawing 5] It is the flow chart which shows a continuation of drawing 4 .

[Drawing 6] It is the flow chart which shows a continuation of drawing 5 .

[Drawing 7] It is the explanatory view showing the example of processing of extracting the difference of an earlier version generator and the program corrected [earlier version].

[Drawing 8] It is the explanatory view showing the example of processing of extracting the difference of an earlier version generator and an upgrade product generator.

[Drawing 9] It is the explanatory view showing the example of program modification processing.

[Description of Notations]

a 102 -- earlier version generator and 103 -- the program corrected [earlier version], a 104 -- upgrade product generator, a 105 -- program automatic generation function, and 106 -- difference -- extract processing (1) and 107 -- difference -- the program corrected [extract processing (2) 108 -- program modification processing and / 109 -- upgrade product].

[Translation done.]

* NOTICES *

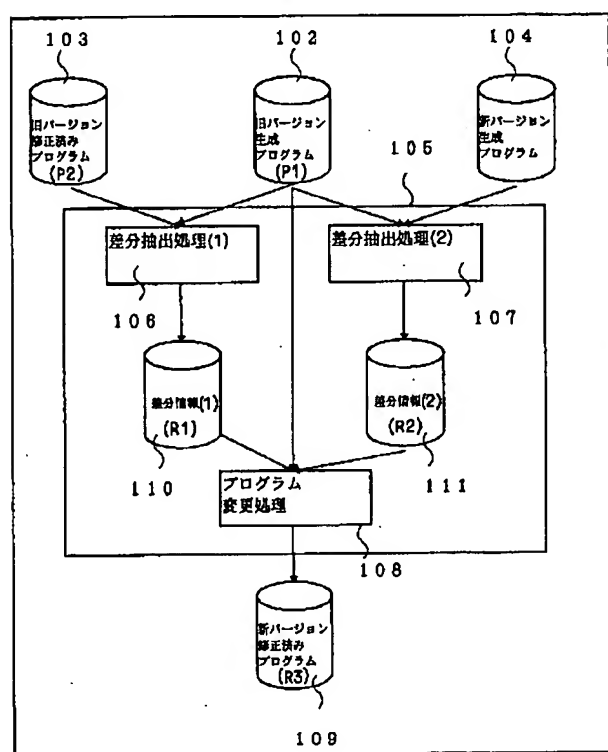
JPO and NCIPI are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DRAWINGS

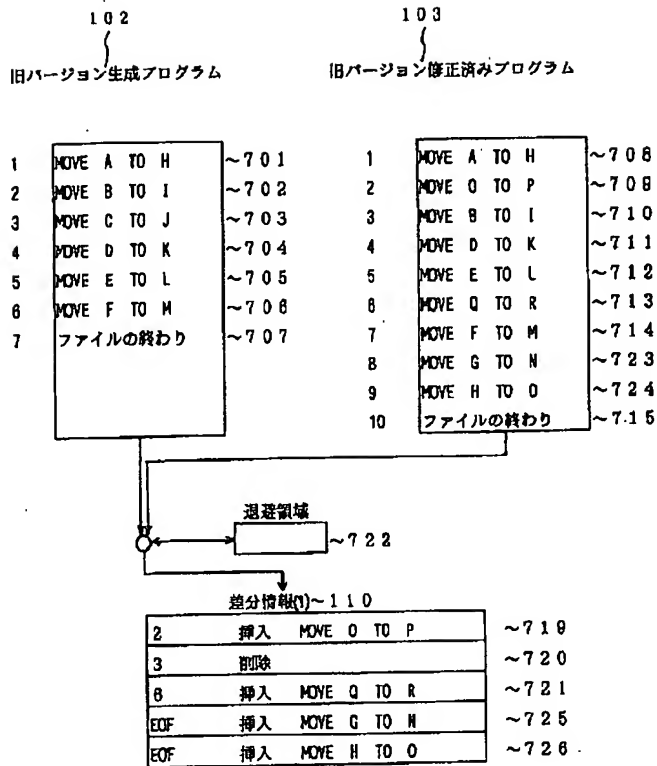
[Drawing 1]

図 1



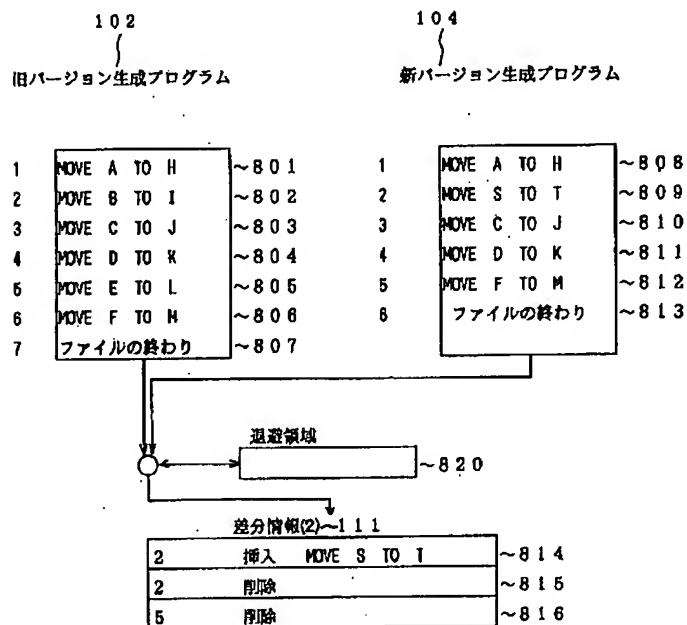
[Drawing 7]

図 7



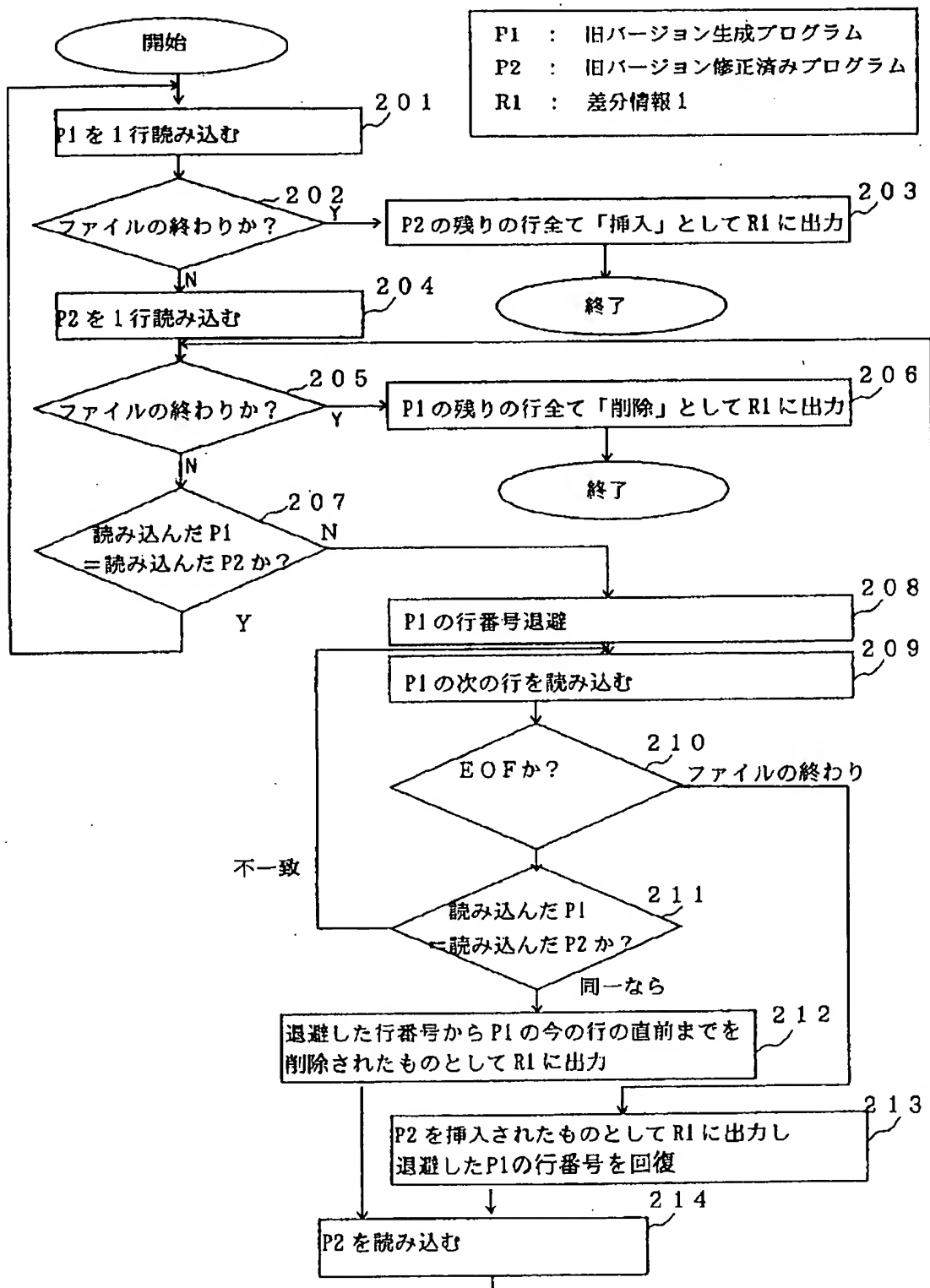
[Drawing 8]

図 8



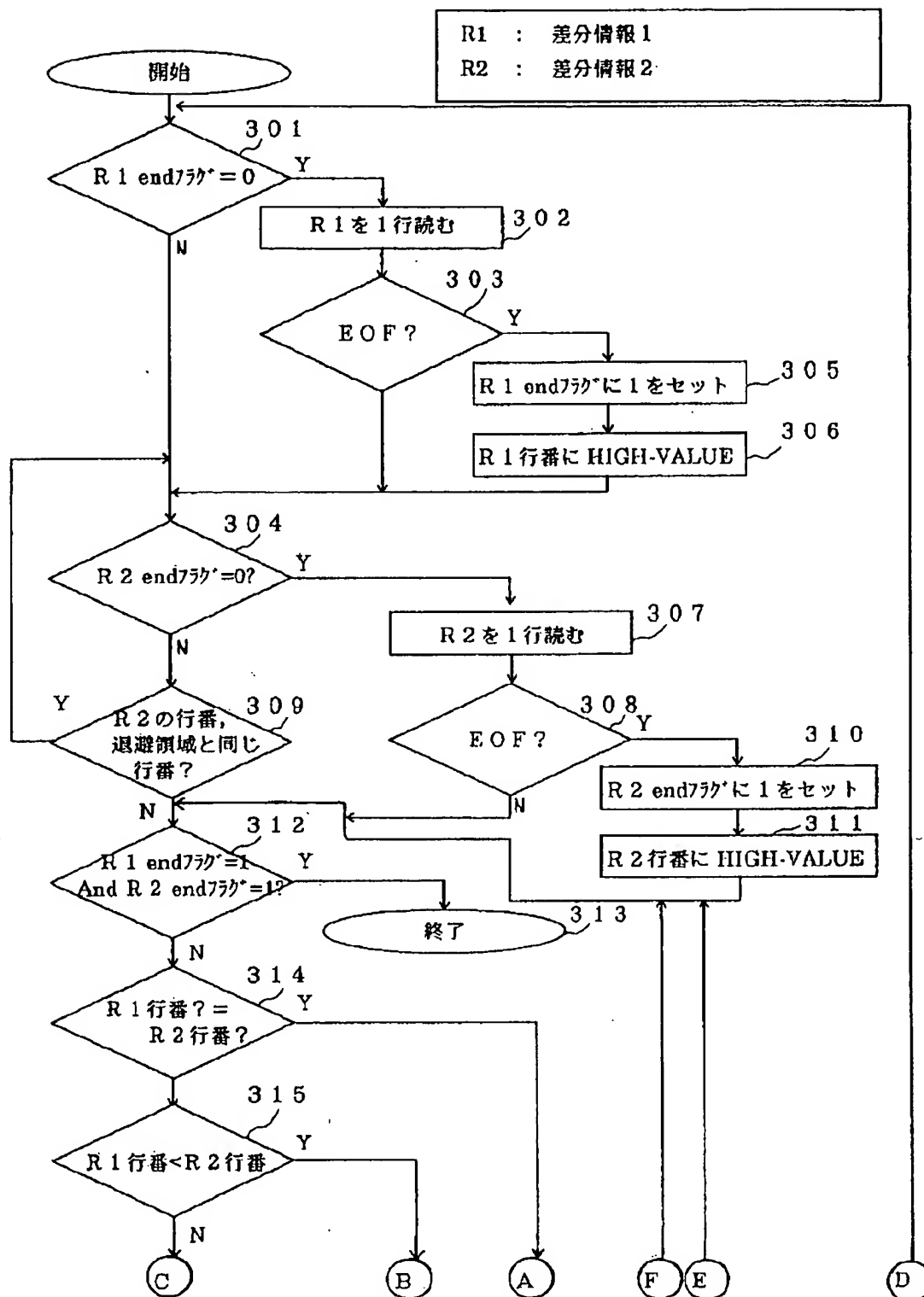
[Drawing 2]

図 2



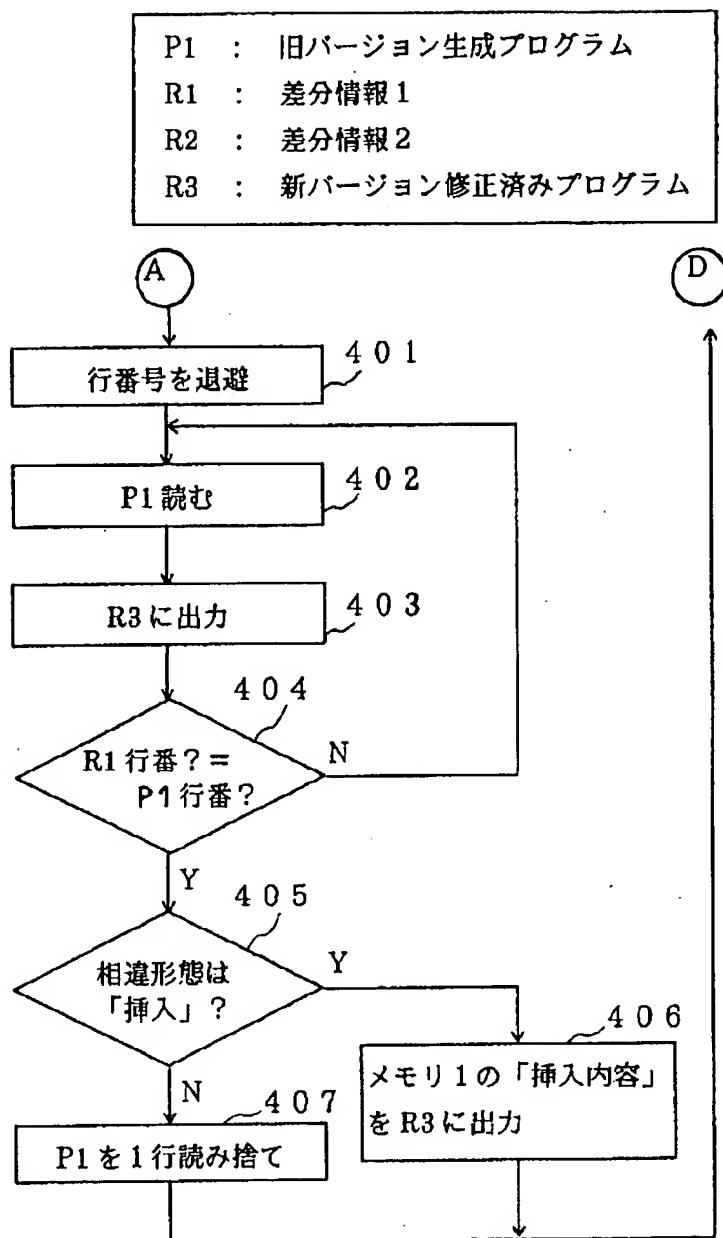
[Drawing 3]

図 3



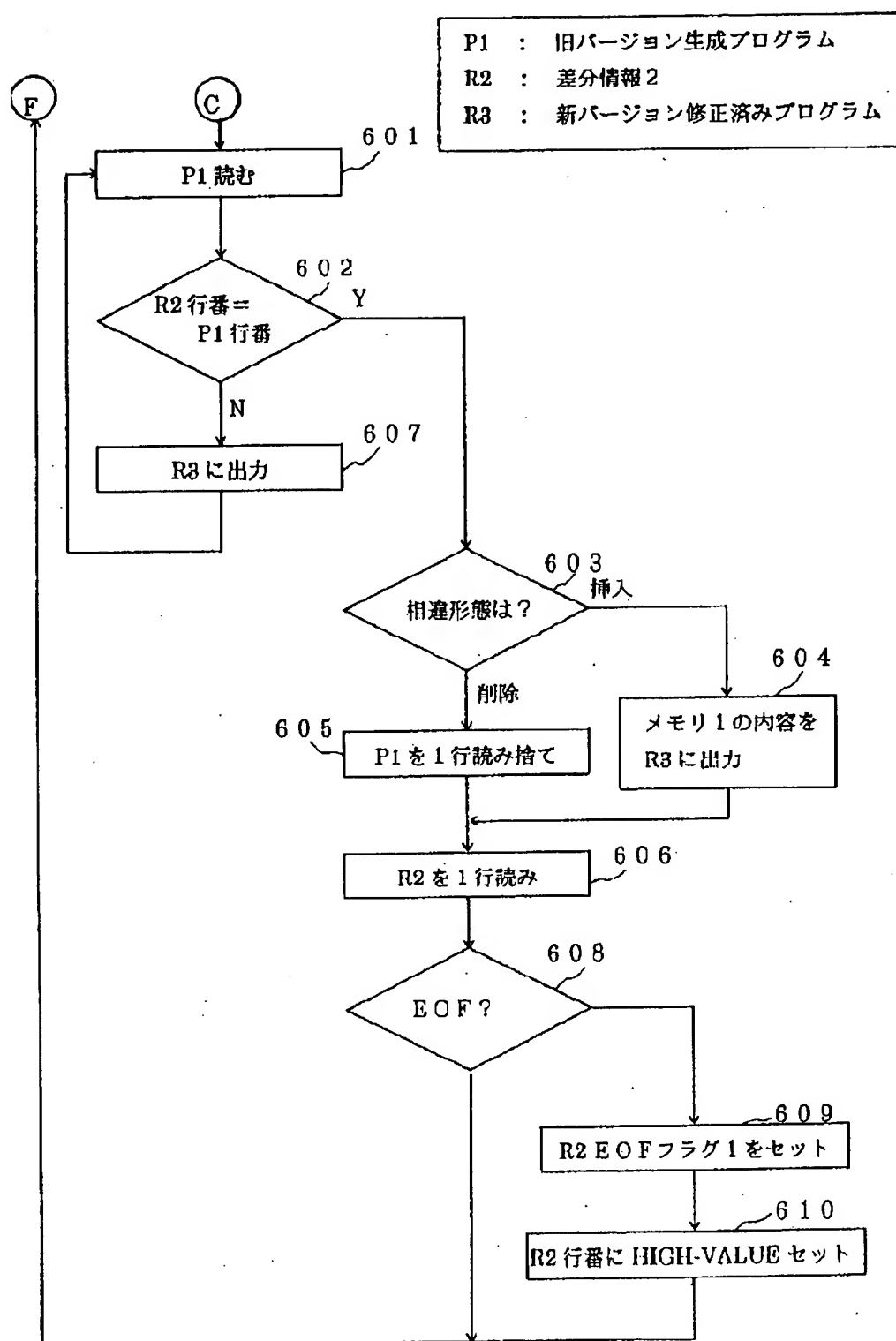
[Drawing 4]

図 4



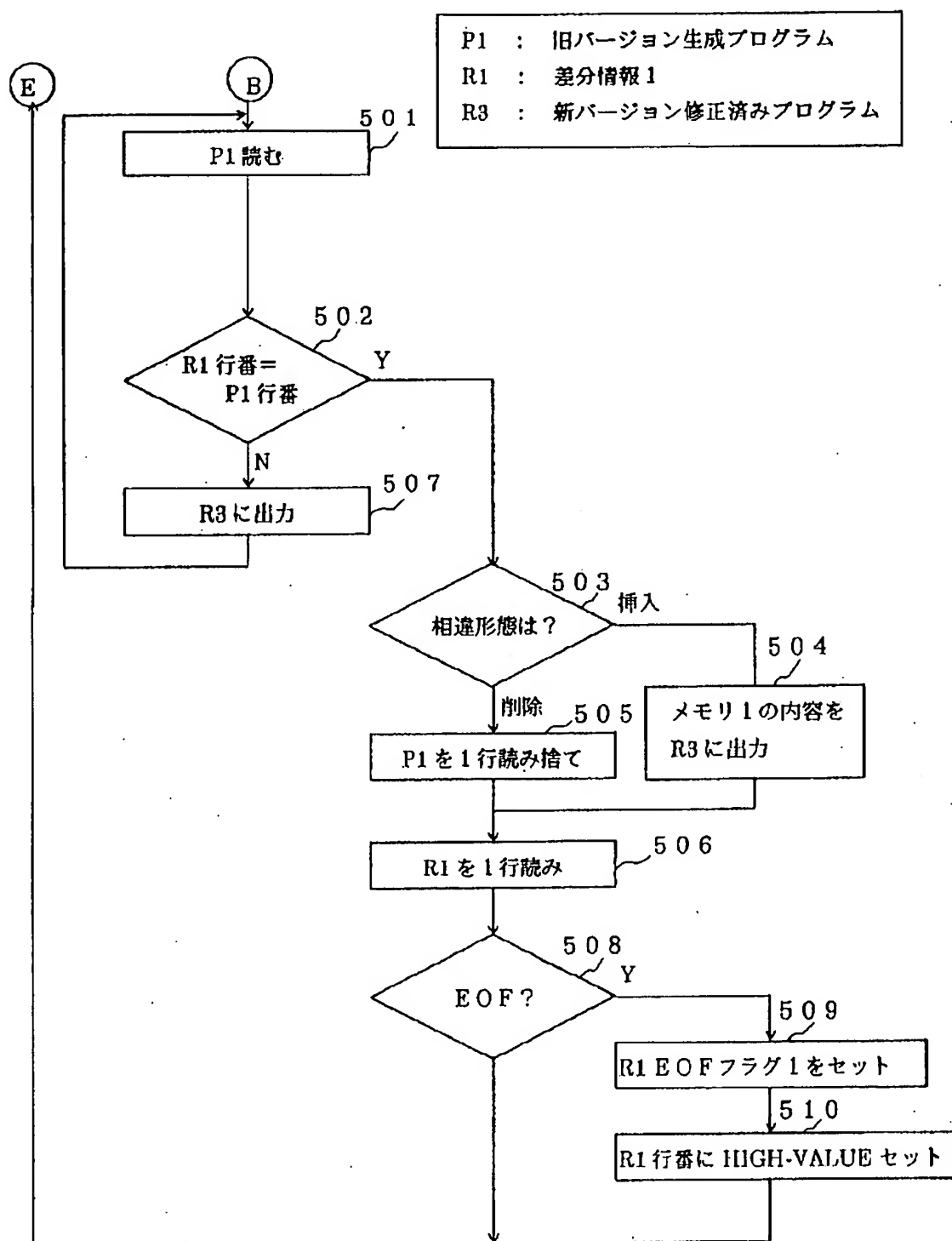
[Drawing 5]

図 5



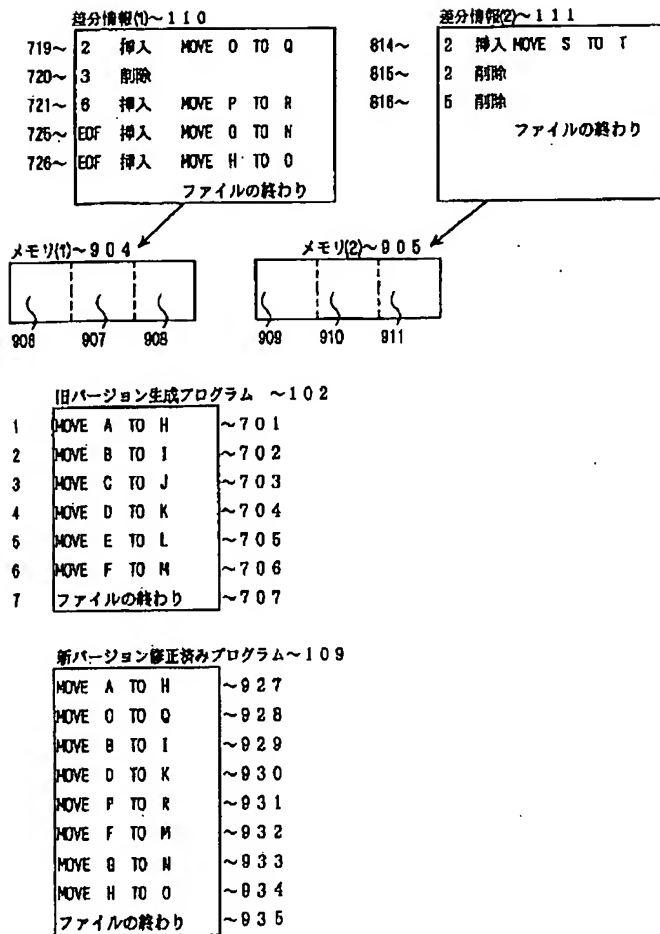
[Drawing 6]

図 6



[Drawing 9]

図 9



[Translation done.]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-21066

(43) 公開日 平成10年(1998) 1月23日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/06	5 4 0		G 0 6 F 9/06	5 4 0 D
				5 4 0 C
	5 3 0			5 3 0 W

審査請求 未請求 請求項の数 1 O L (全 14 頁)

(21) 出願番号 特願平8-173537

(22) 出願日 平成 8 年(1996) 7 月 3 日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目 6 番地

(72) 発明者 川村 正安

東京都江東区新砂一丁目 6 番 27 号 株式会
社日立製作所公共情報事業部内

(72) 発明者 櫻澤 麻美

東京都江東区新砂一丁目 6 番 27 号 株式会
社日立製作所公共情報事業部内

(74) 代理人 弁理士 秋田 収喜

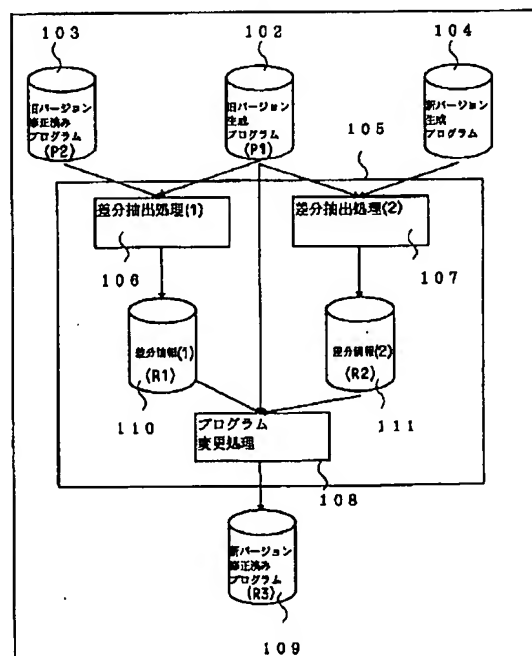
(54) 【発明の名称】 プログラム自動生成方法

(57) 【要約】

【課題】 ソフトウェア生産支援システムのプログラム作成工程において、自動生成したプログラムに対して削除・修正・追加など新たに加えたコーディング部分を保持し、既存の財産をより有効に使用して生成効率を向上させる。

【解決手段】 仕様変更前の旧バージョン生成プログラムと仕様変更後の新バージョン生成プログラムとを比較しその差分を抽出するステップと、旧バージョン生成プログラムに対して追加・修正・削除コーディングを行ったプログラムと旧バージョン生成プログラムとを比較しその差分を抽出するステップと、それぞれの抽出した差分情報から新バージョン仕様情報を用いて生成したプログラムに対し、追加・修正・削除を行うステップとを備える。

図 1



DERWENT-ACC-NO: 1998-150199

DERWENT-WEEK: 199814

COPYRIGHT 2005 DERWENT INFORMATION LTD

TITLE: Automatic program generation
method for software support
system - involves modifying
program corresponding to new
version specification
information obtained based on
extracted differential
information

----- KWIC -----

Basic Abstract Text - ABTX (1):

The method involves comparing an old version
generator before modifying the
specification and a new version generator after
specification modification, to
extract the difference. The new version generator
uses the specification
information of old version generator and forms the
new version specification
information.

Basic Abstract Text - ABTX (2):

The program and old version generator are
compared and the difference is
extracted. Then, the program is modified
corresponding to the new version
specification information obtained based on the
extracted differential

information.

Title - TIX (1):

Automatic program generation method for
software support system - involves
modifying program corresponding to **new version**
specification information
obtained based on extracted differential
information

Standard Title Terms - TTX (1):

AUTOMATIC PROGRAM GENERATE METHOD SOFTWARE
SUPPORT SYSTEM **MODIFIED** PROGRAM
CORRESPOND **NEW VERSION** SPECIFICATION INFORMATION
OBTAIN BASED EXTRACT
DIFFERENTIAL INFORMATION